DETAILED ACTION

1. This communication is responsive to the amendment filed 02/26/2008 and the telephonic interview on 06/18/2008.

Claims 1, 5-10, 12-18, 20, and 21 have been examined and allowed.

2. EXAMINER'S AMENDMENT:

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Authorization for this examiner's amendment was given in a telephone interview with Mr. Serge J. Hodgson (Registration No. 40,017) on 06/18/2008.

The application has been amended as follows:
In the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method comprising:

hooking a critical operating system function;

stalling a call to ~~an~~ the critical operating system function originating from a call module; ~~and~~

determining a location of the call module in a kernel address space of a memory;

determining whether the location ~~said call module~~ is in a driver area of ~~a~~ the kernel address space of ~~a~~ the memory;

determining that said call module is not in said driver area during said determining;

taking protective action to protect a computer system;

providing a notification that said protective action has been taken,

wherein the call module is malicious code that has been injected into a kernel stack/heap through a malicious kernel mode buffer overflow attack.

2.-4 (Canceled)

5. (Currently Amended) The method of Claim [[2]] 1 further comprising terminating said call.

6. (Currently Amended) The method of Claim [[2]] 1 further comprising terminating a parent application comprising said call module.

7. (Currently Amended) The method of Claim [[2]] 1 further comprising determining whether said call module is a known false positive.

8. (Original) The method of Claim 1 further comprising determining that said call module is in said driver area during said determining.

9. (Original) The method of Claim 1 further comprising stalling said call.

10. (Original) The method of Claim 9 further comprising:
determining that said call module is in said driver area during said determining; and
allowing said call to proceed.

11. (Canceled)

12. (Original) The method of Claim 1 further comprising determining if a last mode of operation is a kernel mode.

13. (Original) The method of Claim 1 further comprising disabling loading and unloading of drivers into said kernel address space.

14. (Currently Amended) The method of Claim 13, further comprising, subsequent to said determining whether the location ~~said call module~~ is in a driver area of [[a]] the kernel address space of [[a]] the memory, enabling loading and unloading of said drivers into said kernel address space.

15. (Original) The method of Claim 1 wherein said driver area is static.

16. (Original) The method of Claim 1 wherein said driver area is dynamic.

17. (Original) The method of Claim 16 further comprising keeping said driver area updated as drivers are loaded and unloaded from said kernel address space.

18. (Currently Amended) A method comprising:

hooking driver load and unload functions;

obtaining loaded driver information;

determining a driver area in a kernel address space of a memory; ~~and~~

determining whether a driver has been loaded into or unloaded from said kernel address space, wherein upon a determination that said driver has been loaded into or unloaded from said kernel address space, ~~said method further comprising~~

updating said driver area;

stalling a call to a critical operating system function originating from a call module;

determining whether said call module is in said driver area;

determining that said call module is in said driver area; and

allowing said call to proceed.

19. (Canceled)

20. (Currently Amended) The method of Claim [[19]] 18 wherein said driver area is dynamic.

21. (Currently amended) A computer-program product comprising a tangible computer readable storage medium containing computer code comprising:

a malicious code blocking application for hooking a critical operating system function;

[[a]] said malicious code blocking application for stalling a call to ~~an~~ the critical operating system function originating from a call module; ~~and~~

said malicious code blocking application for determining a location of the call module in a kernel address space of a memory;

said malicious code blocking application further for determining whether the location ~~said call module~~ is in a driver area of ~~a~~ the kernel address space of ~~a~~ the memory;

said malicious code blocking application for determining that said call module is not in said driver area during said determining;

said malicious code blocking application for taking protective action to protect a computer system;

said malicious code blocking application for providing a notification that said protective action has been taken;

wherein the call module is malicious code that has been injected into a kernel stack/heap through a malicious kernel mode buffer overflow attack.

3. **REASONS FOR ALLOWANCE:**

Claims 1, 5-10, 12-18, 20, and 21 are allowed.

The following is an examiner's statement of reasons for allowance:

Applicant's terminal disclaimers have been approved. The prior obviousness type double patenting rejections are withdrawn.

The prior art does not expressly teach or render obvious the invention as recited in independent claims 1, 18, and 21.

The features:

- *"determining that the call module is not in the driver area during the determining; taking protective action to protect a computer system; providing a notification that the protective action has been taken, wherein the call module is malicious code that has been injected into a kernel stack/heap through a malicious kernel mode buffer overflow attack"* (as recited in independent claims 1 and 21); and
- *"stalling a call to a critical operating system function originating from a call module; determining whether the call module is in the driver area; determining that the call module is in the driver area; and allowing the call to proceed"* (as recited in independent claim 18),

when taken in the context of the claims as a whole, were not uncovered in the prior art teachings.

Dependent claims are allowed as they depend upon allowable independent claims.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

CONTACT INFORMATION

Any inquiry concerning this communication or earlier communications from the examiner should be directed to VAN H. NGUYEN whose telephone number is (571) 272-3765. The examiner can normally be reached on Monday-Thursday from 8:30AM - 6:00PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, MENG-AI AN can be reached at (571) 272-3756.

The fax phone number for the organization where this application or proceeding is assigned is **571-273-8300**.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/VAN H NGUYEN/
Primary Examiner, Art Unit 2194